IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 10/051,442	Atty. Dkt. No: 5150-58200
Filed: January 18, 2002	
Inventor(s): Sundeep Chandhoke, Nicolas Vazquez, David W Fuller, and Christopher Cifra	
Title: System and Method for Graphically Creating a Sequence of Motion Control, Machine Vision, and Data Acquisition (DAQ) Operations	
Examiner: Hanne, Sara M.	
Group/Art Unit: 2179	

§
§
§
§
§
§
§
§
§
§
§
§
§
§
§

APPEAL BRIEF

Box: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal filed December 20, 2006 and the Notice of Non-Compliant Appeal Brief mailed July 10, 2007, Appellant presents this amended Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-4, 6-24, 26-30, 32-40, 42-46, 48-54, 57-62, and 66-81 are pending in the application. All of the pending claims stand rejected and are the subject of this appeal. A copy of the claims incorporating entered amendments is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

All amendments have been entered. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE INDEPENDENT CLAIMS

The present claims relate generally to the fields of computer-based motion control, computer-based machine vision, and computer-based data acquisition (DAQ). In particular, the claims relate generally to a system and method enabling a user to graphically create a sequence of motion control operations, machine vision operations, and/or DAQ operations without requiring user programming.

More particularly, independent claim 1 recites a computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality. The method comprises displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The method further comprises creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations selected by the user for inclusion in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The method further comprises storing information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 30 recites a computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ)

functionality. The method comprisesdisplaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The method further comprises creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations selected by the user for inclusion in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The method further comprises performing the specified sequence of operations, where the operations in the sequence implement the motion control, machine vision, and DAQ functionality of the prototype. (*See, e.g., 405 of FIG. 4; p. 23, line 26 – p. 24, line 11*).

Independent claim 36 recites a computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality. The method comprisescreating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The method further comprises recording the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

The operations in the sequence include at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The operations in the sequence are operable to: control a motion control device to move an object (*See, e.g., p. 23, lines 26-30*); control an image acquisition device to acquire one or more images of the object (*See, e.g., p. 24, lines 8-11*); and control a DAQ measurement device to acquire measurement data of the object (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

Independent claim 37 recites a memory medium (*See, e.g., main memory 166 of FIG. 3; floppy disks 104 of FIGs. 2A and 2B; p. 13, lines 24-28*) for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality. The memory medium comprises program instructions executable to display a graphical user interface (GUI) that provides access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The program instructions are further executable to create a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations selected by the user for inclusion in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to

control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The program instructions are further executable to store information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 43 recites a system for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality. The system comprises a processor (*See, e.g., CPU 160 of FIG. 3*); a memory storing program instructions (*See, e.g., main memory 166 of FIG. 3*); and a display device (*See, e.g., display device of computer system 82 in FIGs. 1, 2A, and 2B; video card 180 of FIG. 3*).

The processor is operable to execute the program instructions stored in the memory to display a graphical user interface (GUI) on the display device that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The processor is further operable to execute the program instructions stored in the memory to create a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations selected by the user for inclusion in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The processor is further operable to execute the program instructions stored in the memory to store information representing the sequence of operations in a data structure,

where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 44 recites a system for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality. The system comprises means for displaying a graphical user interface (GUI) that provides GUI access to a set of operations. The set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations. (*See, e.g., computer system 82 of FIGs. 1, 2A, and 2B and main memory 166 of FIG. 3; See also, 801 of FIG. 10; p. 4, lines 15 – 20*).

The system also includes means for creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., computer system 82 of FIGs. 1, 2A, and 2B and main memory 166 of FIG. 3; See also 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The system also includes means for storing information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., computer system 82 of FIGs. 1, 2A, and 2B and main memory 166 of FIG. 3; See also p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 45 recites a computer-implemented method for creating a prototype that includes motion control and machine vision functionality. The method

comprises displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations and one or more machine vision operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The method further comprises creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations selected by the user for inclusion in the sequence includes at least one motion control operation and at least one machine vision operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*).

The method further comprises storing information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 53 recites a computer-implemented method for creating a prototype that includes machine vision and Data Acquisition (DAQ) functionality. The method comprises displaying a graphical user interface (GUI) that provides GUI access to a set of operations, where the set of operations includes one or more machine vision operations and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The method further comprises creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations included in the sequence includes at least one machine vision operation and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-*

28; p. 11, line 28 – p. 12, line 9). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The method further comprises storing information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

Independent claim 61 recites a computer-implemented method for creating a prototype that includes motion control and Data Acquisition (DAQ) functionality. The method comprises displaying a graphical user interface (GUI) that provides GUI access to a set of operations, where the set of operations includes one or more motion control operations and one or more DAQ operations. (*See, e.g., 801 of FIG. 10; p. 4, lines 15 – 20*).

The method further comprises creating a sequence of operations, where creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI. (*See, e.g., 803 of FIG. 10; 421 of FIG. 5; p. 5, lines 27 – 30; p. 27, lines 4-5*). The plurality of operations are included in the sequence without receiving user input specifying program code for performing the plurality of operations. (*See, e.g., p. 5, line 30 – p. 6, line 4*).

The plurality of operations included in the sequence includes at least one motion control operation and at least one DAQ operation. (*See, e.g., p. 4, lines 6-9; p. 5, lines 27-28; p. 11, line 28 – p. 12, line 9*). The at least one DAQ operation included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test. (*See, e.g., p. 6, lines 21-25; p. 17, lines 6-8*).

The method further comprises storing information representing the sequence of operations in a data structure, where the sequence of operations comprises the prototype. (*See, e.g., p. 15, lines 6-7; p. 23, lines 19-25*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Section 102 Rejections

Claims 53, 54, and 57-60 stand rejected under 35 U.S.C. 102(e) as being anticipated by Blowers et al., U.S. Patent No. 6,298,474 (hereinafter "Blowers").

Section 103 Rejections

Claims 1-4, 6-20, 24, 26-30, 32-40, 42-46, 48-52, 61-62, 66-69, 71-73, 76 and 78 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Blowers in view of Weinhofer, U.S. Patent No. 6,442,442 (hereinafter "Weinhofer").

Claims 21-23, 70, 74-75, 77, and 79-81 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blowers in view of Weinhofer and further in view of Wolfson, U.S. Patent No. 6,801,850 (hereinafter "Wolfson").

VII. ARGUMENT

Section 102 Rejections

Claims 53, 54, and 57-60 stand rejected under 35 U.S.C. 102(e) as being anticipated by Blowers et al., U.S. Patent No. 6,298,474 (hereinafter "Blowers"). Appellant respectfully traverses these rejections.

Blowers relates generally to a method and system for interactively developing application software for use in a machine vision system. (See Abstract). Appellant respectfully submits that Blowers does not teach the combination of limitations recited in independent claim 53. In particular, claim 53 recites in pertinent part, "creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI…wherein the plurality of operations included in the sequence includes at least one machine vision operation and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test".

The Examiner has interpreted the "DAQ measurement device" recited in claim 53 as a camera. As argued in detail below, Appellant respectfully submits that this interpretation is improper and is not how those skilled in the art would interpret the term "DAQ measurement device" in light of the specification.

Furthermore, even if the DAQ measurement device is interpreted as a camera, Blowers still does not teach the above-recited claim limitations. The Examiner has interpreted the Caliper tool operation taught by Blowers as the DAQ operation recited in claim 53. With respect to the above-recited claim limitations, the Examiner states on p. 16 of the Office Action of October 20, 2006 that,

> "The camera can acquire data and in combination with software can take measurements. This is clearly shown by Blowers starting Col. 11, line 65 along with the Caliper tool 63, which finds edges used to calculate measurements which can be seen as a DAQ operation. The input images are acquired from the DAQ device, or camera, and then the measurement application takes measurements during image analysis."

However, as noted above, claim 53 recites, "wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test." Blowers' Caliper tool is merely software that executes to detect features in an image that has already been acquired by a camera (Col. 9, lines 44-52). The Caliper tool is not operable to control the camera (which the Examiner has interpreted as the DAQ measurement device), as recited in claim 53. In particular, the Caliper tool is not operable to control the camera to acquire measurement data of a device under test. The Examiner's interpretation of the Caliper tool software measuring aspects of an acquired image as a DAQ operation that controls a DAQ measurement device to acquire measurement data of a device under test is improper, for at least the reasons that the acquired image is not a device under test, and that the Caliper tool does not control the camera at all, much less control the camera to acquire data from a device under test. Thus, Appellant submits that Blowers clearly does not teach the above-recited limitations of claim 53, and thus, claim 53 is patentably distinct over Blowers for at least this reason.

Furthermore, Appellant respectfully submits that the Examiner's interpretation of a DAQ measurement device as a camera is improper. The Examiner states that, "Data Acquisition is a broad term and is taught by the prior art." However, Appellant reminds the Board that claims must be given the broadest reasonable interpretation consistent with the specification. The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. (See MPEP 2111). Appellant respectfully submits that the Examiner's interpretation of a DAQ measurement device as a camera is not consistent with the specification and would not be interpreted as such by those skilled in the art, in light of the specification.

Appellant first notes that the present application explicitly discloses both a camera and a DAQ measurement device and describes them in such a way that it is clear that a DAQ measurement device refers to something other than a camera. For example, Figures 2A and 2B illustrate both a plug-in DAQ board 114 and a camera 132. The specification describes that:

> The host computer 82 may also be coupled to a data acquisition (DAQ) board 114, which may interface through signal conditioning circuitry 124 to the

> UUT. In one embodiment, the signal conditioning circuitry 124 may comprise an SCXI (Signal Conditioning eXtensions for Instrumentation) chassis comprising one or more SCXI modules 126. The sequence developed in the prototyping environment described herein may include one or more DAQ operations. Thus, when the host computer 82 executes the sequence, the DAQ operations may control the DAQ board 114, e.g., to cause the DAQ board 114 to acquire data from the UUT.
>
> Similarly, the sequence may include one or more machine vision operations which cause the host computer 82 to acquire images via the video device or camera 132 and associated image acquisition (or machine vision) card 134. (*p. 17, lines 1-11*)

In light of this description, it is clear that a DAQ measurement device and a camera refer to two different kinds of devices. Thus, Appellant respectfully submits that the Examiner's interpretation of a DAQ measurement device as a camera is not consistent with the specification.

Furthermore, throughout the entire specification, reference is made to three different types of operations: motion control operations, machine vision operations, and data acquisition (DAQ) operations. With respect to machine vision operations, the specification describes that:

> Any of various types of <u>machine vision or image analysis operations</u> may also be provided. Exemplary functions related to machine vision and image analysis include:
>
> · <u>filtering functions for smoothing, edge detection, convolution, etc</u>.
> · morphology functions for modifying the shape of objects in an image, including erosion, dilation, opening, closing, etc.
> · thresholding functions for selecting ranges of pixel values in grayscale and color images
> · particle filtering functions to filter objects based on shape measurements
> · a histogram function that counts the total number of pixels in each grayscale value and graphs it
> · a line profile function that returns the grayscale values of the pixels along a line drawn through the image with a line tool and graphs the values
> · particle analysis functions that computes such measurements on objects in an image as their areas and perimeters
> · a 3D view function that displays an image using an isometric view in which each pixel from the image source is represented as a column of pixels in the 3D view, where the pixel value corresponds to the altitude.
> · <u>an edge detection function that finds edges along a line drawn through the image with a line tool</u>

· a pattern matching function that locates regions of a grayscale image that match a predetermined template
· a shape matching function that searches for the presence of a shape in a binary image and specifies the location of each matching shape
· a **caliper function** that computes measurements such as distances, areas, and angles based on results returned from other image processing functions
· a color matching function that quantifies which colors and how much of each color exist in a region of an image and uses this information to check if another image contains the same colors in the same ratio (*p. 22, line 9 – p. 23, line 5, emphasis added*)

Thus, the specification discloses machine vision operations, including "filtering functions for smoothing, edge detection, convolution, etc." and "an edge detection function that finds edges along a line drawn through the image with a line tool" and "a caliper function that computes measurements such as distances, areas, and angles based on results returned from other image processing functions." Appellant notes that these machine vision operations perform functions very similar to Blowers' Caliper tool. Blowers describes that, "The caliper tool 63 is used to locate pairs of edges within an inspection image. A Region Of Interest (ROI) defines the area to be searched within the image and also the orientation of the edge pairs. The caliper tool 63 is typically used to measure component width by finding edges with sharp contrast changes. The caliper tool 63 generates pass/fail results based on its ability to find edge pairs that are within the specified image." (Col. 9, lines 44-52).

In contrast, the specification describes DAQ operations as follows:

Any of various types of DAQ operations may also be provided. For example, the DAQ operations may include operations related to digital I/O, analog input, analog output, signal conditioning, calibration and configuration (e.g., to calibrate and configure specific devices), counting operations, etc. (*p. 22, lines 5-8)*

Appellant notes that claim 53 recites, "wherein the plurality of operations included in the sequence includes at least one machine vision operation and at least one DAQ operation," thus drawing a distinction between machine vision operations and DAQ operations. In light of the specification, which clearly describes machine vision operations that are very similar to Blowers' Caliper tool, **those skilled in the art would interpret Blower's Caliper tool not as a DAQ operation, but as a machine vision**

operation. Appellant respectfully submits that the Examiner has attempted to reconstruct Appellant's invention by interpreting claim terms in ways that are clearly not consistent with Appellant's specification.

Appellant also notes that Blowers nowhere teaches the use of DAQ operations or a DAQ measurement device, and in fact, the terms "DAQ" and "Data Acquisition" are entirely absent from Blowers' disclosure. Also, FIGS. 2 and 3 show several devices in Blowers' machine vision system, but they do not show a DAQ measurement device. Blowers is directed toward developing software for machine vision applications. DAQ measurement devices are typically used in test and measurement applications, e.g., to acquire measurement data of a device under test. Appellant can find no teaching in Blowers regarding the development of software to perform a measurement application involving measurement data received from a DAQ measurement device. Blowers' machine vision system operates on input images acquired by cameras such as shown in FIG. 2, not on measurement data acquired from a DAQ measurement device.

Thus, for at least the reasons set forth above, Appellant respectfully submits that Blowers does not teach the limitations of "wherein the plurality of operations included in the sequence includes at least one machine vision operation and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test," and thus claim 53, and the claims dependent thereon, are patentably distinct over Blowers for at least this reason. Appellant also respectfully submits that numerous ones of the claims dependent on claim 53 recite further limitations not taught by Blowers, as discussed below.

<u>Claim 58</u>

Claim 58 recites the additional limitations of, "wherein the prototype is operable to: control an image acquisition device to acquire one or more images of the device under test; and control the DAQ measurement device to acquire the measurement data of the device under test." The Office Action cites Col. 11, line 65 et seq. as teaching these features. This portion of Blowers teaches the use of meteorology and image analysis functions to analyze or measure visual features in an acquired image. In other words,

Blowers teaches acquiring images of an object (e.g., images of a device under test) and analyzing or measuring visual features in the acquired image. However, Blowers does not also teach the claimed limitation of, "control the DAQ measurement device to acquire the measurement data of the device under test." Measuring visual features in an acquired image of a device under test is not at all the same as controlling a DAQ measurement device to acquire measurement data of the device under test, as recited in claim 58. In other words, as recited in claim 58 and as well known to those skilled in the art of DAQ test and measurement applications (particularly in light of Applicant's specification as discussed above with respect to claim 53), the DAQ measurement device acquires the measurement data from the device under test itself, not from an image of the device under test.

Blowers does not teach acquiring one or more images of a device under test and also controlling a DAQ measurement device to acquire measurement data of the same device under test. Furthermore, the one or more images in claim 58 are acquired by an image acquisition device, and the measurement data is acquired by a DAQ measurement device. Blowers does not teach the use of two different devices, where one is an image acquisition device to acquire images of a device under test and the other is a DAQ measurement device to acquire measurement data of the device under test.

Furthermore, the Examiner has apparently interpreted the DAQ measurement device as a camera and has interpreted the measurement data as the data acquired by Blowers' image analysis functions. However, claim 58 recites that the measurement data is acquired by the DAQ measurement device. Thus, Blowers' camera acquires the images that are analyzed by the image analysis functions, but the data which the Examiner has equated to the "measurement data" recited in claim 58 is not acquired by the camera (which the Examiner has equated with the DAQ measurement device) but is instead acquired by the image analysis software functions.

Appellant thus respectfully submits that claim 58 is separately patentable over Blowers for at least the reasons set forth above. For similar reasons, Appellant also respectfully submits that Blowers does not teach the additional limitations recited in claim 57 of,

wherein the prototype is operable to:

> acquire one or more images of the device under test;
> analyze the acquired images of the device under test; and
> acquire the measurement data of the device under test.

As discussed above, Blowers teaches acquired one or more images of an object (e.g., images of a device under test) and analyzing the acquired images, but does not teach the additional limitation of acquiring measurement data <u>of the device under test</u>.

<u>Claim 60</u>

Claim 60 recites the additional limitations of, "<u>automatically</u> generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, wherein automatically generating the graphical program comprises <u>automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes</u>." The Office Action refers to the tree structure such as shown in FIG. 7 of Blowers. However, this tree structure is not <u>automatically</u> generated. On the contrary, the icons are included in the tree structure <u>in response to user input selecting the icons</u> (Col. 8, lines 61 – 67). Blowers does not teach the concept of automatically generating a graphical program, wherein automatically generating the graphical program comprises automatically including a plurality of interconnected nodes in the graphical program <u>without user input specifying the nodes</u>, as recited in claim 60.

With respect to the above arguments, the Examiner states on p. 17 of the Office Action of October 20, 2006 that, "Col. 3, line 64 of Blowers states "generate/"teach" a machine vision computer program without having to write any code". In such a manner the graphical program can be automatically generated without the user specifying the specific nodes once it has be taught. However, what Blowers actually states is the following:

> The benefits accruing to the method and system of the present invention are numerous. For example, the method and system:
>
> 1) **<u>Interactively</u>** generate/"teach" a machine vision computer program without having to write any code.

Regarding these features, Blowers states in the Abstract:

> "A method, a system and a computer-readable storage medium having stored therein a program for interactively developing a graphical control-flow structure and associated application software for use in a machine vision system is provided. The structure is a tree view structure including a control sequence having at least one node."

Thus, Blowers clearly teaches that the user interactively develops the tree view structure, where the nodes/icons are included in the tree view structure in response to user input (see Abstract; Col. 3, lines 60-65; Col. 8, lines 61-67). Thus, the nodes/icons in the tree view structure (which the Examiner has equated with the nodes recited in claim 60) are not automatically included in the tree view structure without user input specifying the nodes/icons.

Appellant thus respectfully submits that claim 60 is separately patentable over Blowers for at least the reasons set forth above.

Section 103 Rejections

Claims 1-4, 6-20, 24, 26-30, 32-40, 42-46, 48-52, 61-62, 66-69, 71-73, 76 and 78 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Blowers in view of Weinhofer, U.S. Patent No. 6,442,442 (hereinafter "Weinhofer"). Appellant respectfully traverses these rejections.

As per claims 1-4, 6-20, 24, 26-30, 32-40, 42-44 and 61-62, 66-69, 71-73, 76 and 78, the independent claims recite similar limitations regarding DAQ operations as discussed above with respect to claim 53, and so the arguments presented above with respect to DAQ operations and DAQ devices regarding Blowers apply with equal force to these claims. Nor does Weinhofer remedy these deficiencies of Blowers, since Weinhofer also fails to teach or suggest the claimed limitations regarding DAQ operations. More specifically, regarding the independent claims 1, 30, 36, 37, 43, 44, and 61, neither Blowers nor Weinhofer teaches or suggests "wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test", as recited in these claims.

Furthermore, independent claims 1, 30, 36, 37, 43, 44, 45, and 61 also recite other limitations that are not by Blowers or Weinhofer, taken either singly or in combination. For example, claim 1 recites as follows:

> 1. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the method comprising:
>
> displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations;
>
> creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises <u>including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations</u>;
>
> wherein the plurality of operations included in the sequence includes <u>at least one motion control operation</u>, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;
>
> wherein the method further comprises storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype. (*Emphasis added*)

Thus, the method of claim 1 comprises, in pertinent part, including at least one motion control operation in the sequence <u>without specifying program code for performing the motion control operation(s)</u>. Appellant respectfully submits that Weinhofer does not teach these limitations.

Weinhofer teaches that the user creates a <u>graphical data flow program</u> that comprises a plurality of interconnected icons, where connections between the icons represent data flow between the icons, and where the icons are implemented in an object-oriented programming environment (Col. 3, line 63 – Col. 4, line 7; and Col. 6, lines 5 – 38). The icons and the connections between the icons constitute graphical <u>program code</u> that defines the functionality of the graphical program. Weinhofer does not teach including motion control operations in a sequence in response to user input, <u>without</u> receiving user input specifying program code for performing the motion control operations.

Furthermore, Appellant respectfully submits that Weinhofer does not combine with Blowers to form a case of *prima facie* obviousness with respect to the above claim limitations. In particular, Appellant respectfully submits that there is no evidence of any teaching, suggestion, or motivation to combine Blowers and Weinhofer. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999) (Emphasis added).

The Examiner asserts on p. 6 of the Office Action of October 20, 2006 that, "One would have been motivated to make such a combination because an all-purpose graphical automotive controller would have been obtained, as taught by Weinhofer." However, as Appellant has pointed out in the responses to the Office Actions, Appellant can find no teaching in Weinhofer of this alleged "all-purpose graphical automotive controller." In three separate responses to Office Actions (see the responses filed on July 5, 2005, December 20, 2005, and July 24, 2006), Appellant requested the Examiner to cite where Weinhofer teaches this "all-purpose graphical automotive controller", but the Examiner was unresponsive to this request. Appellant respectfully submits that Weinhofer does not teach an "all-purpose graphical automotive controller", and furthermore, even if this were disclosed it still would not amount to a clear and particular teaching or suggestion for combining Blowers with Weinhofer, as required to form a *prima facie* case of obviousness.

Appellant submits that neither Weinhofer nor Blowers contain any clear teaching or suggestion for combining the two references. Blowers is directed toward developing software for machine vision applications, e.g., for analyzing images. Weinhofer is directed toward developing software for motion control applications, e.g., for moving devices. Appellant can find no teaching in Blowers regarding the development of software to perform an application involving motion control operations as taught in

Weinhofer nor any teaching regarding a need to perform motion control operations such as taught in Weinhofer. Similarly, Appellant can find no teaching in Weinhofer regarding the development of software to perform an application involving machine vision operations as taught in Blowers nor any teaching regarding a need to perform machine vision operations such as taught in Blowers. Thus, there is no motivation or suggestion for incorporating the machine vision operations taught in Blowers into Weinhofer's system or for incorporating the motion control operations taught in Weinhofer into Blowers's system.

In response to Appellant's arguments that there is no clear teaching or suggestion for combining Blowers and Weinhofer, the Examiner asserts on p. 18 of the Office Action of October 20, 2006 that, "In this case, Weinhofer explains how motion controllers are part of many industrial control systems including programmable controller systems (Col. 1, line 48) and therefore can be interpreted as a control system for many different purposes and that it would be advantageous to use Weinhofer along with such other systems. Blowers teaches such a system in it's programmable controller system."

However, the cited portion of Weinhofer states that, "Motion controllers may for example be provided in the form of modules for a programmable controller system or as PC-based expansion cards or stand-alone units that communicate with the programmable controller system via a network communication link." Appellant submits that this simply describes a motion control system architecture in which motion controllers are provided in the form of modules for a programmable controller system. This says nothing about performing a machine vision application such as taught in Blowers and does not amount to a clear and particular teaching or suggestion for combining Weinhofer's motion control operations with Blowers' system.

In response to Appellant's arguments, the Examiner also cites Col. 2, line 57 – Col. 3, line 25, where Weinhofer describes that existing programming interfaces do not enable the relationship between various motion control axes to be readily ascertained. To solve this problem, Weinhofer teaches at Col. 4, lines 8-20 that:

> Advantageously, the programming interface according to the preferred embodiment of the invention explicitly indicates the physical relationship between the various motion control axes. <u>The various motion control axes are represented by icons, and the icons are connected with connection lines that</u>

> represent data flow between the motion control axes. Additional icons are provided that show relationships such as gearing, position cams, time cams, and so on. The programming interface is thus organized based on the physical relationship between the axes, and the physical relationships for the entire system are displayed to the user in a single workspace, without the user having to click on numerous icons. (*Emphasis added*)

However, Blowers teaches at Col. 2, lines 47-53 that:

> An object of the present invention is to provide a method and system for interactively developing application software for use in a machine vision system and computer-readable storage medium having a program for executing the method wherein the user teaches an imaging programming task without writing any code. Consequently, the user need not be a programmer. (*Emphasis added*)

Thus, Blowers emphasizes that the user is able to develop application software for use in a machine vision system without writing any code, and thus, the user need not be a programmer. In contrast, Weinhofer teaches that **the user creates a program** that comprises a plurality of interconnected icons (Col. 3, line 63 – Col. 4, line 7; and Col. 6, lines 5 – 38). In creating the program, the user is necessarily a programmer, and the user necessarily writes program code for the program, where the program code is in the form of icons and data flow lines connecting the icons.

One would not be motivated to combine Weinhofer with Blowers because Weinhofer emphasizes the importance of the interconnected icons in the graphical data flow program, since the icons indicate the physical relationship between the various motion control axes, and also show relationships such as gearing, position cams, time cams, etc. Creating a graphical data flow program by specifying graphical code comprising icons and nodes is an important aspect of Weinhofer's teaching, since the interconnected icons in the graphical data flow program illustrates the physical relationship between the various motion control axes. Thus, Weinhofer actually teaches away from any combination with Blowers, since Blowers teaches developing application software for a machine vision system without the user being a programmer and without the user writing program code.

Thus, for at least the reasons set forth above, Appellant respectfully submits that Weinhofer does not combine with Blowers to form a case of *prima facie* obviousness for claim 1, and thus, claim 1 is patentably distinct over the cited references. Inasmuch as

independent claims 30, 36, 37, 43, 44, 45, and 61 recite similar limitations as those discussed above with respect to claim 1, Appellant submits that the other independent claims are also patentably distinct over the cited references, for reasons similar to those discussed above.

Since the independent claims are patentably distinct, Appellant respectfully submits that the dependent claims are also patentably distinct, for at least this reason. Appellant further submits that numerous ones of the dependent claims recite further distinctions not taught by the cited references, as discussed below.

Claims 9 and 42

Claim 9 recites similar limitations as discussed above with respect to claim 58. In particular, claim 9 recites the limitations of, "wherein the prototype is operable to:… control an image acquisition device to acquire one or more images of the device under test; and control the DAQ measurement device to acquire the measurement data of the device under test."

As discussed above with respect to claim 58, Blowers fails to teach these limitations. Nor does Weinhofer remedy these deficiencies of Blowers. Appellant thus respectfully submits that claim 9 is separately patentable over the cited references for at least these reasons. Inasmuch as claim 42 recites similar limitations as claim 9, Appellant respectfully submits that claim 42 is also patentable over the cited references, for reasons similar to those discussed above.

Claim 72

Claim 72 recites the additional limitations of, "wherein said controlling the DAQ measurement device to acquire the measurement data of the device under test comprises controlling the DAQ measurement device to acquire waveform data of the device under test." The Examiner has interpreted the DAQ measurement device as Blowers' camera and has interpreted the measurement data as the data acquired by Blowers' image analysis software functions. As argued above, Appellant submits that these interpretations are improper. Furthermore, it is well known that a camera acquires images and does not acquire waveform data as recited in claim 72. Also, Blowers

nowhere teaches or even remotely suggests the concept of controlling a DAQ measurement device to acquire waveform data of a device under test, either in the portions cited by the Examiner or elsewhere.

Appellant thus respectfully submits that claim 72 is separately patentable over the cited references for at least the reasons set forth above.

Claim 15

Claim 14 recites in pertinent part, "displaying a graphical panel including graphical user interface elements for setting properties of the first operation". Claim 15 depends on claim 14 and recites the further limitation of, "wherein the graphical panel is automatically displayed in response to including the first operation in the sequence." In the rejection of claim 15, the Examiner cites Blowers as teaching this limitation at Col. 9, line 7 et seq. and Col. 12, lines 8-10. However, Appellant can find nothing either here or elsewhere in Blowers, nor in Weinhofer, regarding automatically displaying a graphical panel for setting properties of an operation in response to including the operation in a sequence.

Appellant thus respectfully submits that claim 15 is separately patentable over the cited references for at least the reasons set forth above.

Claim 16

Claim 14 recites in pertinent part, "displaying a graphical panel including graphical user interface elements for setting properties of the first operation". Claim 16 depends on claim 14 and recites the further limitations of, "receiving user input requesting to configure the first operation" and "displaying the graphical panel for configuring the first operation in response to the request." In the rejection of claim 16, the Examiner cites Blowers as teaching these limitations at Col. 8, line 61 et seq. However, Appellant can find nothing either here or elsewhere in Blowers, nor in Weinhofer, regarding displaying a graphical panel for setting properties of an operation in a sequence in response to user input requesting to configure the operation.

Appellant thus respectfully submits that claim 16 is separately patentable over the cited references for at least the reasons set forth above.

Claims 24, 52, and 68

Claim 24 recites similar limitations as discussed above with respect to claim 60. In particular, claim 24 recites the further limitations of, "automatically generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein automatically generating the graphical program comprises automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes."

As discussed above with respect to claim 60, Blowers fails to teach these limitations. Nor does Weinhofer remedy these deficiencies of Blowers. Appellant thus respectfully submits that claim 24 is separately patentable over the cited references for at least these reasons. Inasmuch as claims 52 and 68 recites similar limitations as claim 24, Appellant respectfully submits that claims 52 and 68 are also patentable over the cited references, for reasons similar to those discussed above.

Claim 26

Claim 26 recites the additional limitations of, "wherein the graphical program comprises a graphical data flow program, wherein the plurality of interconnected nodes visually indicates data flow that occurs among the nodes." The Examiner asserts that these limitations are taught by Blowers, citing Col. 3, lines 14-35, Col. 11, line 15, and Col. 8, line 49 et seq. However, Appellant can find no teaching either here or elsewhere in Blowers that the tree structure (which the Examiner has interpreted as the graphical program) is a graphical data flow program or that the interconnected nodes in the tree structure visually indicates data flow that occurs among the nodes.

As well known in the art of graphical programming, the term "graphical data flow program" refers to a graphical program where the nodes are connected according to a data flow format, e.g., such that nodes in the graphical program are interconnected by lines that visually indicate data flow among the nodes. For example, a line or wire from an output terminal of a first node to an input terminal of a second node may serve as a

visual indication that output data produced by the first node is passed as input data to the second node.

Appellant respectfully submits that Blowers' tree structure is not a graphical data flow program and does not visually indicate data flow among the nodes in the tree. In FIG. 7, Blowers illustrates an example tree structure. Appellant respectfully submits that data flow among the nodes in the tree structure is not clear from the tree hierarchy. For any given node in the tree, it is not clear whether the node receives data from any of the other nodes in the tree, and if so, which one or ones. Furthermore, a tree hierarchy such as shown in FIG. 7 is a fairly rigid structure that does not allow data flow among the nodes to be uniquely represented. For example, suppose that in one case the "Alignment" node only passes data directly to the "Template" node and not to the "Blob" node. Suppose that in another case the "Alignment" node passes data directly to both the "Template" node and to the "Blob" node. In both of these cases, a tree structure such as taught in Blowers would have exactly the same visual appearance. Blowers does not teach the concept of a graphical data flow program suitable for visually indicating the particular data flow that occurs among a plurality of nodes. Also, Blowers nowhere describes that the interconnected nodes in the tree structure visually indicate data flow that occurs among the nodes, as recited in claim 26.

Appellant thus respectfully submits that claim 26 is separately patentable over the cited references for at least the reasons set forth above.

<u>Claim 27</u>

Claim 27 recites the additional limitations of, "<u>automatically generating a text-based program based on the sequence of operations</u>, wherein the text-based program is executable to perform the specified sequence of operations, <u>wherein the text-based program comprises a plurality of lines of textual source code</u>, wherein automatically generating the text-based program comprises <u>automatically including the lines of textual source code in the text-based program without user input specifying the lines of textual source code</u>."

The Examiner asserts that these limitations are taught by Blowers, citing Col. 3, lines 15-45 and Col. 8, line 61 et seq. However, Appellant can find no teaching either

here or elsewhere in Blowers regarding automatically generating a text-based program based on a sequence of operations, in combination with the other limitations recited in claim 27.

Appellant thus respectfully submits that claim 27 is separately patentable over the cited references for at least the reasons set forth above.

Claim 28

Claim 28 recites the additional limitations of:

> wherein said displaying the GUI comprises a first application displaying the GUI;
>
> wherein said creating the sequence comprises the first application creating the sequence;
>
> wherein the method further comprises:
>
> the first application receiving a request to invoke execution of the sequence from a second program external to the first application; and
>
> the first application executing the sequence of operations in response to the request from the second program, wherein the first application executing the sequence comprises the first application invoking execution of software routines to perform the plurality of operations in the sequence. (*Emphasis added*)

In the rejection of claim 28 the Examiner cites FIG. 3 of Blowers. However, at Col. 8, lines 28-40, Blowers describes FIG. 3 as follows:

> "Referring now to FIG. 3, there is illustrated in block diagram form various software and hardware components for interactively developing a graphical, control-flow structure and associated application software for use in the machine vision system 20 of FIG. 2 using the computer system of FIG. 1 without the need for a user to write any code. The method and system of the present invention "marry" the ActiveX-COM standard with the commonly used "tree view" method of navigation and hierarchy. The system allows for adding new machine vision functions. The design includes a method of controlling the flow of sequences based on conditional branches and forced routing changes."

Appellant respectfully submits that the various software components illustrated in FIG. 3 are all part of a single application for interactively developing a graphical, control-flow structure and associated application software for use in the machine vision system 20, as described by Blowers above. Blowers nowhere teaches the limitation recited in claim 28 of, "the first application receiving a request to invoke execution of the sequence from a second program external to the first application".

Appellant thus respectfully submits that claim 28 is separately patentable over the cited references for at least the reasons set forth above.

Claim 29

Claim 29 recites the additional limitations of:

> automatically converting the sequence of operations to a hardware configuration format usable for configuring configurable hardware of an embedded device to perform the sequence of operations; and
>
> configuring the configurable hardware of the embedded device to perform the sequence of operations using the hardware configuration format.

The Examiner asserts that these limitations are taught by Blowers, citing Col. 2, line 47 et seq. The cited portion of Blowers relates generally to Blowers' method and system for interactively developing application software for use in a machine vision system. As described above, Blower's system allows the user to interactively create a tree structure including various machine vision operations. However, Blowers nowhere teaches anything at all regarding automatically converting the tree structure to a hardware configuration format usable for configuring configurable hardware of an embedded device to perform the sequence of operations or configuring the configurable hardware of the embedded device to perform the sequence of operations using the hardware configuration format, as recited in claim 29.

Appellant thus respectfully submits that claim 29 is separately patentable over the cited references for at least the reasons set forth above.

Claims 69, 73, 76, and 78

Claim 69 recites the additional limitations of, "displaying a visual indication of results of performing the sequence while the sequence is being created, wherein the visual indication enables a user to evaluate the results of performing the sequence…"

The Examiner asserts that these limitations are taught by Blowers, citing Col. 4, lines 46 et seq. The cited portion of Blowers relates generally to Blowers' method and system for interactively developing application software for use in a machine vision system. As described above, Blower's system allows the user to interactively create a tree structure including various machine vision operations. The tree structure provides a

visual indication of which machine vision operations have been included in the tree structure. However, the tree structure provides no indication at all of the <u>results of performing the sequence</u>. Appellant can find no teaching in the cited portion of Blowers regarding displaying a visual indication of results of performing the sequence while the sequence is being created, as recited in claim 69.

Furthermore, Appellant can find no teaching in the cited portions of Blowers regarding the additional limitations in claim 69 of, "wherein interactively displaying the visual indication comprises: for each operation in at least a subset of the operations included in the sequence, <u>updating the displayed visual indication in response to including the operation in the sequence</u> in order to visually indicate a change in the results of performing the sequence, wherein the change is caused by including the operation in the sequence, wherein updating the displayed visual indication provides interactive visual feedback to the user indicating the change caused by including the operation in the sequence."

Appellant thus respectfully submits that claim 69 is separately patentable over the cited references for at least the reasons set forth above. Inasmuch as claims 73, 76, and 78 recite similar limitations as claim 69, Appellant respectfully submits that these claims are also patentable over the cited references, for reasons similar to those discussed above.

<u>Claim 71</u>

Claim 71 recites the additional limitations of:

> automatically converting the sequence of operations to a hardware configuration format usable for configuring a Field Programmable Gate Array (FPGA) device to perform the sequence of operations; and
>
> configuring the FPGA device to perform the sequence of operations using the hardware configuration format.

In the rejection of claim 71, the Examiner states:

> Blowers and Weinhofer fails to explicitly teach configuring the FPGA device to perform the sequence of operations using the hardware configuration format. as recited in the claims. Within the field of the invention, it would be obvious to one of ordinary skill in the art to program a removable device like a FPGA. One would have been motivated to make such a combination because a removable hardware device for executing sequenced instructions would have been obtained.

However, Appellant respectfully reminds the Board that, "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)" as stated in the MPEP §2142 (*emphasis added*).

The Examiner has not provided any reference that teaches the above-recited features of automatically converting the sequence of operations to a hardware configuration format usable for configuring a Field Programmable Gate Array (FPGA) device to perform the sequence of operations and configuring the FPGA device to perform the sequence of operations using the hardware configuration format. Nor has the Examiner provided any evidence of a teaching or suggestion in the prior art to produce the combination of features recited in claim 71.

Appellant respectfully submits that the art does not fairly teach or suggest to one to make the specific combination as claimed in claim 71 and that the Examiner has simply engaged in improper hindsight analysis in the rejection of claim 71.

Appellant thus respectfully submits that claim 71 is separately patentable over the cited references for at least the reasons set forth above.

<u>Claims 21-23, 70, 74-75, 77, and 79-81</u>

Dependent claims 21-23, 70, 74-75, 77, and 79-81 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blowers in view of Weinhofer and further in view of Wolfson, U.S. Patent No. 6,801,850 (hereinafter "Wolfson"). Appellant respectfully traverses these rejections.

Applicant first respectfully reminds the Board that if an independent claim is non-obvious under 35 U.S.C. 103, then any claim depending therefrom is non-obvious. In re

Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Applicant thus respectfully submits that since the independent claims have been shown above to be patentably distinct and non-obvious over the prior art, these dependent claims are also patentably distinct and non-obvious, for at least this reason.

Appellant also respectfully reminds the Board that, "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." *In re Oetiker*, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). See also *In re Deminski*, 796 F.2d 436, 230 USPQ 313 (Fed. Cir. 1986); *In re Clay*, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992).

Wolfson relates generally to tracking moving objects (see Abstract). For example, Wolfson teaches that it may be desirable to track the location of a moving object such as a cell phone, personal digital assistant, laptop, etc. and predict its future movement. (See Col. 1, line 25 – Col. 2, line 62).

In contrast, the limitations recited in claims 21-23, 70, 74-75, 77, and 79-81 relate to the field of computer-based motion control. As described in the Description of the Related Art of the present application,

> Computer-based motion control involves precisely controlling the movement of a device or system. Computer-based motion control is widely used in many different types of applications, including applications in the fields of industrial automation, process control, test and measurement automation, robotics, and integrated machine vision, among others. A typical computer-based motion system includes components such as the moving mechanical device(s), a motor with feedback and motion I/O, a motor drive unit, a motion controller, and software to interact with the motion controller. (*p. 1, lines 7-14*)

Wolfson is entirely unrelated to the field of computer-based motion control and thus is not in the field of Applicant's endeavor. Wolfson teaches a system for tracking the location of an object such as a cell phone, but teaches nothing at all about controlling the location of an object through a computer-based motion control system. Appellant respectfully submits that Wolfson pertains to an entirely different problem than the particular problem with which the inventors in the present application were concerned and that Wolfson is not reasonably pertinent to the subject matter recited in claims 21-23,

70, 74-75, 77, and 79-81. Appellant thus submits that it is improper to rely on Wolfson as a basis for rejection of these claims.

Furthermore, Appellant also submits that Wolfson, taken either singly or in combination with the other references, does not teach the limitations recited in claims 21-23, 70, 74-75, 77, and 79-81. For example, claim 21 recites:

> 21. (Previously Presented) The method of claim 1, wherein the sequence includes two or more motion control operations, and wherein the method further comprises:
>
> displaying a graph illustrating a spatial trajectory cumulatively performed by the two or more motion control operations, wherein the graph provides a visual preview of the spatial trajectory cumulatively performed by the two or more motion control operations.

In the rejection of claim 21 the Examiner cites FIG. 4 of Wolfson. Wolfson teaches that,

> "FIG. 4 illustrates a two-dimensional uncertainty 162 associated with a trajectory 156. Three points 150, 152, 154 generate the trajectory. The continuous trajectory is the solid line, indicated generally by 162. <u>The moving object is anticipated to travel along the trajectory</u> 156 from the first point 150 to the second point 152 and then to the third point 154."

Thus FIG. 4 illustrates a trajectory along which a self-moving object (or an object carried by a person or automobile for example) is anticipated to travel but does not illustrate a spatial trajectory cumulatively performed by <u>two or more motion control operations</u>, wherein the graph provides a visual preview of the spatial trajectory cumulatively performed by the two or more motion control operations.

Furthermore, Appellant respectfully submits that FIG. 4 is merely a drawing to aid the reader of the Wolfson patent in understanding Wolfson's invention. Wolfson does not teach any method comprising displaying a graph as recited in claim 21, where the graph provides a visual preview of a spatial trajectory cumulatively performed by two or more motion control operations that have been included in a sequence in response to user input.

Appellant thus respectfully submits that claim 21 is separately patentable over the cited references for at least the reasons set forth above.

Claim 22 recites the further limitation that the displayed graph of claim 21 "comprises a two-dimensional graph illustrating a two-dimensional display of the spatial trajectory performed by the two or more motion control operations." In the rejection of claim 22 the Examiner cites FIG. 8 of Wolfson. However, Wolfson describes FIG. 8 as follows:

> FIG. 8 illustrates a traffic incident model. The traffic incident shows two phases, the persistent interval 350 and the recovery interval 352. The build-up interval is not shown. In the persistent interval 350 the traffic speed is a constant value, shown by line 354. During the recovery interval 352 the speed recovers to the normal speed at a constant rate. Dashed line 358 depicts the normal speed, while line 356 shows the linearly increase speed during the recovery interval 352. In addition to the traffic incident model depicted in FIG. 7, there are two specialized traffic incidents.

Thus, FIG. 8 illustrates a traffic incident model and is not a 2-D version of FIG. 4 as apparently asserted by the Examiner. Furthermore, FIG. 8 does not illustrate a two-dimensional display of a spatial trajectory performed by two or more motion control operations that have been included in a sequence in response to user input, as recited in claim 22.

Appellant thus respectfully submits that claim 22 is separately patentable over the cited references for at least the reasons set forth above.

Claim 70 recites the further limitations of:

> wherein the plurality of operations included in the sequence includes a plurality of motion control operations;
>
> wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:
>
> for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

Apellant notes that the Office Action does not address these limitations and provides no reasoning for the rejection of claim 70. Appellant respectfully submits that

the cited references do not fairly teach or suggest these limitations. In particular, there is no teaching or suggestion to combine the references to obtain the features of, "updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence."

Appellant thus respectfully submits that claim 70 is separately patentable over the cited references for at least the reasons set forth above. Claims 74, 75, 77, 79, 80, and 81 recite similar limitations as claim 70, which are also not addressed by the Office Action. Appellant respectfully submits that these claims are also patentably distinct over the cited references, for reasons similar to those discussed above.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-4, 6-24, 26-30, 32-40, 42-46, 48-54, 57-62, and 66-81 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-58200/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2007-08-10 JCH/JLB

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments, and as on appeal.

1. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the method comprising:

displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations;

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the method further comprises storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

2. (Previously Presented) The method of claim 1, further comprising:

accessing the data structure to determine the plurality of operations in the sequence;

determining software routines to execute in order to perform the plurality of operations in the sequence; and

executing the software routines.

3. (Previously Presented) The method of claim 1, further comprising:

receiving user input specifying a first parameter value for a first operation in the sequence;

wherein said storing the information representing the sequence of operations in the data structure comprises storing the first parameter value in the data structure;

wherein the method further comprises executing software routines corresponding to operations in the sequence, wherein executing the software routines comprises executing a first software routine corresponding to the first operation, wherein said executing the first software routine comprises passing the first parameter value to the first software routine.

4. (Previously Presented) The method of claim 1,

wherein the information representing the sequence of operations in the data structure does not comprise program code.

5. (Canceled)

6. (Previously Presented) The method of claim 1,

wherein the machine vision operation(s) in the sequence are operable to perform one or more of:

acquire one or more images of the device under test; and/or

analyze one or more acquired images of the device under test.

7. (Previously Presented) The method of claim 1,

wherein the motion control operation(s) in the sequence are operable to control a motion control device in order to move the device under test.

8. (Previously Presented) The method of claim 1,

wherein the machine vision operation(s) in the sequence are operable to analyze one or more acquired images of the device under test; and

wherein the motion control operation(s) in the sequence are operable to control a motion control device in order to move the device under test.

9. (Previously Presented) The method of claim 1,

wherein the prototype is operable to:

control a motion control device to move the device under test;

control an image acquisition device to acquire one or more images of the device under test; and

control the DAQ measurement device to acquire the measurement data of the device under test.

10. (Previously Presented) The method of claim 1, further comprising:

performing the sequence of operations;

wherein said performing the sequence of operations comprises executing software routines in order to perform each operation in the sequence.

11. (Previously Presented) The method of claim 1, further comprising:

creating program instructions executable to perform the sequence of operations; and

executing the program instructions.

12. (Previously Presented) The method of claim 1, further comprising:

configuring a first operation in the sequence in response to user input specifying configuration information for the first operation, wherein configuring the first operation changes a function performed by the first operation; and

displaying information in response to the user input specifying the configuration information in order to visually indicate the change in the function performed by the first operation.

13. (Previously Presented) The method of claim 12,

wherein the user input specifying the configuration information for the first operation does not include user input specifying program code.

14. (Previously Presented) The method of claim 12, further comprising:

displaying a graphical panel including graphical user interface elements for setting properties of the first operation, wherein the user input specifying the configuration information for the first operation comprises user input to the graphical panel to set one or more properties of the first operation.

15. (Previously Presented) The method of claim 14,

wherein the graphical panel is automatically displayed in response to including the first operation in the sequence.

16. (Previously Presented) The method of claim 14, further comprising:

receiving user input requesting to configure the first operation; and

displaying the graphical panel for configuring the first operation in response to the request.

17. (Previously Presented) The method of claim 1,

wherein the GUI includes an area which visually represents the operations in the sequence;

wherein the method further comprises:

for each operation included in the sequence, updating the area visually representing the operations in the sequence to illustrate the included operation, in response to the user input selecting the operation from the GUI.

18. (Previously Presented) The method of claim 17,

wherein the area visually representing the operations in the sequence displays a plurality of icons, wherein each icon visually indicates one of the operations in the sequence;

wherein said updating the area visually representing the operations in the sequence to illustrate the included operation comprises displaying a new icon to visually indicate the included operation.

19. (Previously Presented) The method of claim 1,

wherein the GUI displays a plurality of buttons, wherein each button corresponds to a particular operation and is operable to add the operation to the sequence in response to user input selecting the button;

wherein said including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence in response to user input selecting corresponding buttons from the plurality of buttons.

20. (Previously Presented) The method of claim 1,

wherein the one or more motion control operations in the set of operations to which the GUI provides GUI access include:

a straight line move operation;

an arc move operation; and

a contoured move operation.

21. (Previously Presented) The method of claim 1, wherein the sequence includes two or more motion control operations, and wherein the method further comprises:

displaying a graph illustrating a spatial trajectory cumulatively performed by the two or more motion control operations, wherein the graph provides a visual preview of the spatial trajectory cumulatively performed by the two or more motion control operations.

22. (Previously Presented) The method of claim 21,

wherein the displayed graph comprises a two-dimensional graph illustrating a two-dimensional display of the spatial trajectory performed by the two or more motion control operations.

23. (Previously Presented) The method of claim 21,

wherein the displayed graph comprises a three-dimensional graph illustrating a three-dimensional display of the spatial trajectory performed by the two or more motion control operations.

24. (Previously Presented) The method of claim 1, further comprising:

automatically generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein automatically generating the graphical program comprises automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes.

25. (Canceled)

26. (Previously Presented) The method of claim 24,

wherein the graphical program comprises a graphical data flow program, wherein the plurality of interconnected nodes visually indicates data flow that occurs among the nodes.

27. (Previously Presented) The method of claim 1, further comprising:

automatically generating a text-based program based on the sequence of operations, wherein the text-based program is executable to perform the specified sequence of operations, wherein the text-based program comprises a plurality of lines of textual source code, wherein automatically generating the text-based program comprises automatically including the lines of textual source code in the text-based program without user input specifying the lines of textual source code.

28. (Previously Presented) The method of claim 1,

wherein said displaying the GUI comprises a first application displaying the GUI;

wherein said creating the sequence comprises the first application creating the sequence;

wherein the method further comprises:

the first application receiving a request to invoke execution of the sequence from a second program external to the first application; and

the first application executing the sequence of operations in response to the request from the second program, wherein the first application executing the sequence comprises the first application invoking execution of software routines to perform the plurality of operations in the sequence.

29. (Previously Presented) The method of claim 1, further comprising:

automatically converting the sequence of operations to a hardware configuration format usable for configuring configurable hardware of an embedded device to perform the sequence of operations; and

configuring the configurable hardware of the embedded device to perform the sequence of operations using the hardware configuration format.

30. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the method comprising:

displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations;

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in

the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the method further comprises performing the specified sequence of operations, wherein the operations in the sequence implement the motion control, machine vision, and DAQ functionality of the prototype.

31. (Canceled)

32. (Previously Presented) The method of claim 30, further comprising:

storing information representing the sequence of operations in a data structure.

33. (Previously Presented) The method of claim 32,

wherein the information representing the sequence of operations in the data structure does not comprise program code.

34. (Previously Presented) The method of claim 32, wherein said performing the sequence of operations comprises:

accessing the data structure to determine the plurality of operations in the sequence;

determining software routines to execute in order to perform the plurality of operations in the sequence; and

executing the software routines.

35. (Previously Presented) The method of claim 32, further comprising:

receiving user input specifying a first parameter value for a first operation in the sequence;

wherein said storing the information representing the sequence of operations in the data structure comprises storing the first parameter value in the data structure;

wherein said performing the sequence of operations comprises executing software routines corresponding to operations in the sequence, wherein executing the software routines comprises executing a first software routine corresponding to the first operation, wherein said executing the first software routine comprises passing the first parameter value to the first software routine.

36. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the method comprising:

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the method further comprises recording the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype;

wherein the operations in the sequence include at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein the operations in the sequence are operable to:

control a motion control device to move an object;

control an image acquisition device to acquire one or more images of the object; and

control a DAQ measurement device to acquire measurement data of the object.

37. (Previously Presented) A memory medium for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the memory medium comprising program instructions executable to:

display a graphical user interface (GUI) that provides access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations;

create a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the program instructions are further executable to store information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

38. (Previously Presented) The memory medium of claim 37, wherein the program instructions are further executable to:

access the data structure to determine the plurality of operations in the sequence;

determine software routines to execute in order to perform the plurality of operations in the sequence; and

execute the software routines.

39. (Previously Presented) The memory medium of claim 37, wherein the program instructions are further executable to:

receive user input specifying a first parameter value for a first operation in the sequence;

wherein said storing the information representing the sequence of operations in the data structure comprises storing the first parameter value in the data structure;

wherein the program instructions are further executable to execute software routines corresponding to operations in the sequence, wherein executing the software routines comprises executing a first software routine corresponding to the first operation, wherein said executing the first software routine comprises passing the first parameter value to the first software routine.

40. (Previously Presented) The memory medium of claim 37,

wherein the information representing the sequence of operations in the data structure does not comprise program code.

41. (Canceled)

42. (Previously Presented) The memory medium of claim 37,

wherein the prototype is operable to:

control <u>a</u> motion control device to move the device under test;

analyze one or more acquired images of the device under test; and

control the DAQ measurement device to acquire the measurement data of the device under test.

43. (Previously Presented) A system for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the system comprising:

a processor;

a memory storing program instructions;

a display device;

wherein the processor is operable to execute the program instructions stored in the memory to:

display a graphical user interface (GUI) on the display device that provides GUI access to a set of operations, wherein the set of operations includes one or more motion

control operations, one or more machine vision operations, and one or more DAQ operations;

create a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the processor is operable to further execute the program instructions stored in the memory to store information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

44. (Previously Presented) A system for creating a prototype that includes motion control, machine vision, and Data Acquisition (DAQ) functionality, the system comprising:

means for displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations, one or more machine vision operations, and one or more DAQ operations;

means for creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation, at least one machine vision operation, and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the system further comprises means for storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

45. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control and machine vision functionality, the method comprising:

displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations and one or more machine vision operations;

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation and at least one machine vision operation;

wherein the method further comprises storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

46. (Previously Presented) The method of claim 45, further comprising:

accessing the data structure to determine the plurality of operations in the sequence;

determining software routines to execute in order to perform the plurality of operations in the sequence; and

executing the software routines.

47. (Canceled)

48. (Previously Presented) The method of claim 45,

wherein the prototype is operable to perform one or more of:

control motion of a device;

acquire images; and

analyze the acquired images.

49. (Previously Presented) The method of claim 45,

wherein the prototype is operable to:

control motion of a device;

acquire images; and

analyze the acquired images.

50. (Previously Presented) The method of claim 45,

wherein the prototype is operable to:

control a motion control device to move an object; and

control an image acquisition device to acquire one or more images of the object.

51. (Previously Presented) The method of claim 45, further comprising:

performing the sequence of operations;

wherein said performing the sequence of operations comprises executing software routines in order to perform each operation in the sequence.

52. (Previously Presented) The method of claim 45, further comprising:

automatically generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, wherein automatically generating the graphical program comprises automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes.

53. (Previously Presented) A computer-implemented method for creating a prototype that includes machine vision and Data Acquisition (DAQ) functionality, the method comprising:

displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more machine vision operations and one or more DAQ operations;

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one machine vision operation and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the method further comprises storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

54. (Previously Presented) The method of claim 53, further comprising:

accessing the data structure to determine the plurality of operations in the sequence;

determining software routines to execute in order to perform the plurality of operations in the sequence; and

executing the software routines.

55. - 56. (Canceled)

57. (Previously Presented) The method of claim 53,

wherein the prototype is operable to:

acquire one or more images of the device under test;

analyze the acquired images of the device under test; and

acquire the measurement data of the device under test.

58. (Previously Presented) The method of claim 53,

wherein the prototype is operable to:

control an image acquisition device to acquire one or more images of the device under test; and

control the DAQ measurement device to acquire the measurement data of the device under test.

59. (Previously Presented) The method of claim 53, further comprising:

performing the sequence of operations;

wherein said performing the sequence of operations comprises executing software routines in order to perform each operation in the sequence.

60. (Previously Presented) The method of claim 53, further comprising:

automatically generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, wherein automatically generating the graphical program comprises automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes.

61. (Previously Presented) A computer-implemented method for creating a prototype that includes motion control and Data Acquisition (DAQ) functionality, the method comprising:

displaying a graphical user interface (GUI) that provides GUI access to a set of operations, wherein the set of operations includes one or more motion control operations and one or more DAQ operations;

creating a sequence of operations, wherein creating the sequence comprises including a plurality of operations in the sequence in response to user input selecting each operation in the plurality of operations from the GUI, wherein including the plurality of operations in the sequence in response to the user input selecting each operation in the plurality of operations from the GUI comprises including the plurality of operations in the sequence without receiving user input specifying program code for performing the plurality of operations;

wherein the plurality of operations included in the sequence includes at least one motion control operation and at least one DAQ operation, wherein at least one of the DAQ operations included in the sequence is operable to control a DAQ measurement device to acquire measurement data of a device under test;

wherein the method further comprises storing information representing the sequence of operations in a data structure, wherein the sequence of operations comprises the prototype.

62. (Previously Presented) The method of claim 61, further comprising:

accessing the data structure to determine the plurality of operations in the sequence;

determining software routines to execute in order to perform the plurality of operations in the sequence; and

executing the software routines.

63. - 65. (Canceled)

66. (Previously Presented) The method of claim 61,

wherein the prototype is operable to:

control a motion control device to move the device under test; and

control the DAQ measurement device to acquire the measurement data of the device under test.

67. (Previously Presented) The method of claim 61, further comprising:

performing the sequence of operations;

wherein said performing the sequence of operations comprises executing software routines in order to perform each operation in the sequence.

68. (Previously Presented) The method of claim 61, further comprising:

automatically generating a graphical program based on the sequence of operations, wherein the graphical program is executable to perform the specified sequence of operations, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, wherein automatically generating the graphical program comprises automatically including the plurality of interconnected nodes in the graphical program without user input specifying the nodes.

69. (Previously Presented) The method of claim 1, further comprising:

displaying a visual indication of results of performing the sequence while the sequence is being created, wherein the visual indication enables a user to evaluate the results of performing the sequence, wherein interactively displaying the visual indication comprises:

for each operation in at least a subset of the operations included in the sequence, updating the displayed visual indication in response to including the operation in the sequence in order to visually indicate a change in the results of performing the sequence, wherein the change is caused by including the operation in the sequence, wherein updating the displayed visual indication provides interactive visual feedback to the user indicating the change caused by including the operation in the sequence.

70. (Previously Presented) The method of claim 1,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

71. (Previously Presented) The method of claim 1, further comprising:

automatically converting the sequence of operations to a hardware configuration format usable for configuring a Field Programmable Gate Array (FPGA) device to perform the sequence of operations; and

configuring the FPGA device to perform the sequence of operations using the hardware configuration format.

72. (Previously Presented) The method of claim 1,

wherein said controlling the DAQ measurement device to acquire the measurement data of the device under test comprises controlling the DAQ measurement device to acquire waveform data of the device under test.

73. (Previously Presented) The method of claim 30, further comprising:

displaying a visual indication of results of performing the sequence while the sequence is being created, wherein the visual indication enables a user to evaluate the

results of performing the sequence, wherein interactively displaying the visual indication comprises:

for each operation in at least a subset of the operations included in the sequence, updating the displayed visual indication in response to including the operation in the sequence in order to visually indicate a change in the results of performing the sequence, wherein the change is caused by including the operation in the sequence, wherein updating the displayed visual indication provides interactive visual feedback to the user indicating the change caused by including the operation in the sequence.

74. (Previously Presented) The method of claim 30,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

75. (Previously Presented) The method of claim 36,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the

motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

76. (Previously Presented) The memory medium of claim 37, wherein the program instructions are further executable to:

display a visual indication of results of performing the sequence while the sequence is being created, wherein the visual indication enables a user to evaluate the results of performing the sequence, wherein interactively displaying the visual indication comprises:

for each operation in at least a subset of the operations included in the sequence, updating the displayed visual indication in response to including the operation in the sequence in order to visually indicate a change in the results of performing the sequence, wherein the change is caused by including the operation in the sequence, wherein updating the displayed visual indication provides interactive visual feedback to the user indicating the change caused by including the operation in the sequence.

77. (Previously Presented) The memory medium of claim 37,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the program instructions are further executable to interactively display a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive

visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

78. (Previously Presented) The system of claim 43, wherein the processor is operable to further execute the program instructions stored in the memory to:

display a visual indication of results of performing the sequence while the sequence is being created, wherein the visual indication enables a user to evaluate the results of performing the sequence, wherein interactively displaying the visual indication comprises:

for each operation in at least a subset of the operations included in the sequence, updating the displayed visual indication in response to including the operation in the sequence in order to visually indicate a change in the results of performing the sequence, wherein the change is caused by including the operation in the sequence, wherein updating the displayed visual indication provides interactive visual feedback to the user indicating the change caused by including the operation in the sequence.

79. (Previously Presented) The system of claim 43,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the processor is operable to further execute the program instructions stored in the memory to interactively display a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

80. (Previously Presented) The method of claim 45,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

81. (Previously Presented) The method of claim 61,

wherein the plurality of operations included in the sequence includes a plurality of motion control operations;

wherein the method further comprises interactively displaying a graph illustrating a spatial trajectory cumulatively performed by the plurality of motion control operations, wherein interactively displaying the graph comprises:

for each motion control operation in the plurality of motion control operations included in the sequence, updating the graph in response to including the motion control operation in the sequence in order to visually indicate a change in the spatial trajectory, wherein the change in the spatial trajectory is caused by including the motion control operation in the sequence, wherein updating the graph provides interactive visual feedback to the user indicating the change caused by including the motion control operation in the sequence.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.